UNIVERSITÀ
DI TRENTO

Service Design and Engineering 2021

Final Project

StarMap

A star gazing advisor

Shuxin
Zheng

Academic Year 2021/2022

Contents

Overview

This project is mainly designed as a star gazing advisor. It first ask for the geographical position of the user, then analyse the light pollution level, sunrise and sunset time to give some advise for star gazing. It will also give a star map according to the location message, and user can also customize some parameters, for example: only show the stars lighter than 6.0, etc.
By defined like beyond, I use the APIs listed here:

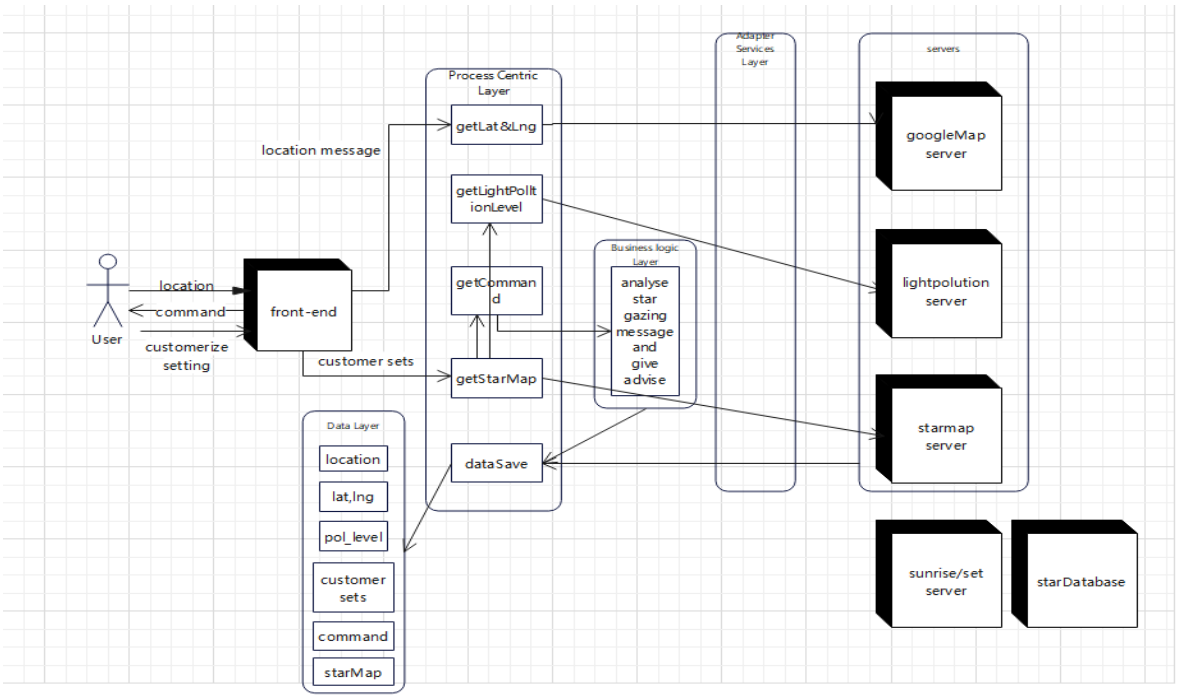
Location: GoogleMap API

Light pollution map: RESTFUL API - https://www.lightpollutionmap.info/

Sky map: RESTFUL API - https://www.fourmilab.ch/

Sunrise/sunset time: RESTFUL API - self-developed calculation service

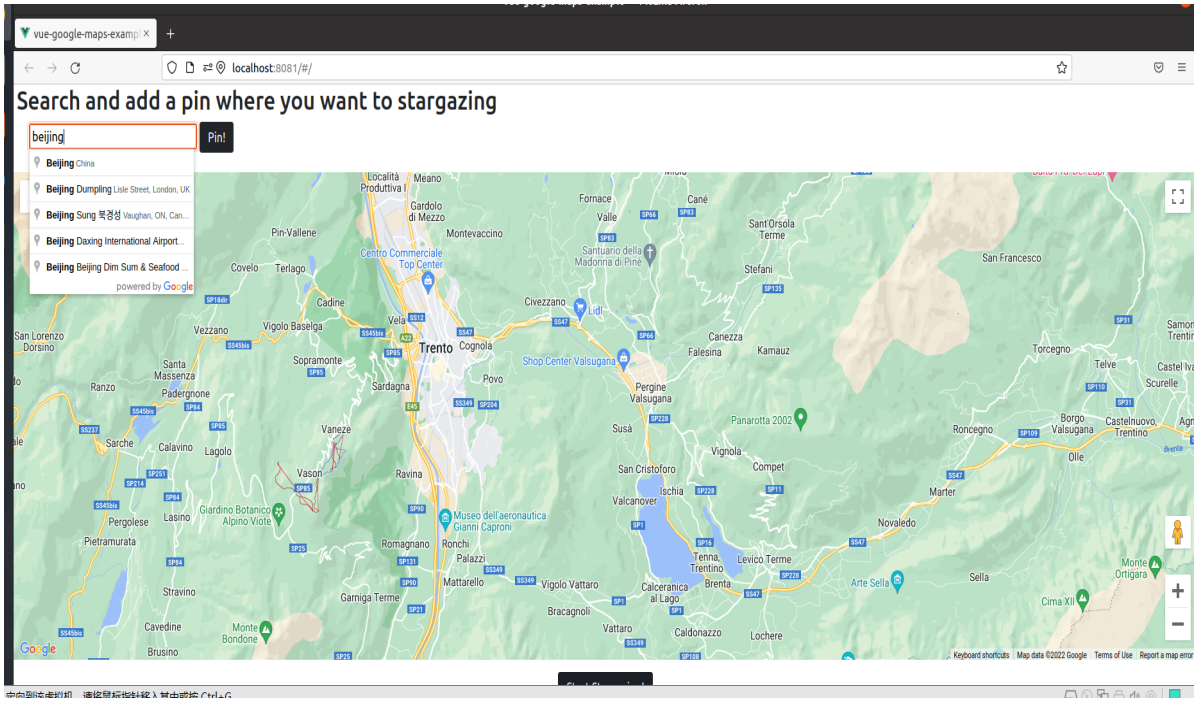
There be four layers and one server(sunrise/sunset calculation server) in this project,and all code is valid in the github: https://github.com/lannnnn/SkyMap. But due to the technology Vue used in front-end, the data layer is not run a really important position in this project - though I have finished and fully tested all the functions for parameters setter and getter, due to the limitation of one-person's work, I didn't finish the identification part. So in current developing state it will lead to ungraceful code when comparing to directly use the object tree. Therefore I comment out these part of code, it's surely useful! It maintains data using file operations, and the structure will surely be valid to supporting a multi-user framework. Anyway, I will specifically share the details for the four layers and one servers in this report. And here I put the graph as the structure overview(in order not to make the picture confusing, I leave out some of the servers):



1 Workflow

To use the SkyMap service, users can first chose the position and than pin it for star your star gazing!

After chosen your location, you will get the information about the there - the longitude, latitude, the light pollution level around there. Based on these messages, the StarMap will give a map which is customizable together with some analyse and advise such like these area is suitable or not to star-gazing, and which



magnitude is the visible star. A sunset and sunrise time forecast is also valid to give rough reference about the gazing time. Here should be a preview of the page:

Users can custom the star map and try to find the visible stars according to the given suggestions. Besides, I integrated a star database link here as the blue text. By open this link, users can search the related messages by using the star names show in the star map. I didn't collecting specific messages in this project because I think it's better for stargazers to get more complete messages rather than a filtered one.

So given the formal name should be perfect!

2 Layers

Here I would like to talk about the layers more specifically. All the layers are build using nodejs, and some specific packages was used based on requirements.

Data Layer There be many different kinds of technologies that can be used in this layer, for instance, database, file operations etc. I used the file operations here for the user data is not yet very large and also it's more flexible than a database which the structure is fixed. The file is organized in format of JSON like shown below. This layer would be valid for multi-user one if we add the user identification messages in the file structure or even directly by name the files according to the userId.

```
{
    "sunrise":"4:01",
    "sunset":"18:25",
    "lightPol":48.8,
    "comment":"Not a suitable stargazing ...",
    "message":"The light pollution here within 5km is pretty heavy..."
}
```

Adapter Layer This layer mainly process the message like abstracting/combining the sending payload also the return value to the specific format. For example, build the RANGE information according to the given location which will be used on light pollution API to find the minimum light pollution levels nearby.

Business logic Layer Business logic Layer in this project mainly works as the decision maker. It takes the information from the center process layer through API and give back the suggestions and some messages about the star gazing.

Process Centric Layer I made this layers combined with the front-end, mainly process the requirements from users and distribute the tasks to different layers and give back the result. This process make the starMap works as a interactive software.

3 Server

Also a brief introduce for the sunrise/sunset service. This service is build for give the sunrise and sunset time at the given location, and the results is given on UTC time. I didn't find a current exiting service which is suitable for my project so I build it myself by referring some scientist methods.

This service is build using nodejs, and the only exposed API take the latitude and longitude information as input and return the sunrise/sunset UTC time in string format, which will also be shown in the API document.